Spectrally Queued Feature Selection for Robotic Visual Odometery

David M. Pirozzo, Philip A. Frederick, Shawn Hunt, Bernard Theisen, Mike Del Rose
TARDEC, 6501 E.11 Mile Rd., Warren, MI, USA 48397-5000

ABSTRACT

Over the last two decades, research in Unmanned Vehicles (UV) has rapidly progressed and become more influenced by the field of biological sciences. Researchers have been investigating mechanical aspects of varying species to improve UV air and ground intrinsic mobility, they have been exploring the computational aspects of the brain for the development of pattern recognition and decision algorithms and they have been exploring perception capabilities of numerous animals and insects. This paper describes a 3 month exploratory applied research effort performed at the US ARMY Research, Development and Engineering Command's (RDECOM) Tank Automotive Research, Development and Engineering Center (TARDEC) in the area of biologically inspired spectrally augmented feature selection for robotic visual odometry. The motivation for this applied research was to develop a feasibility analysis on multi-spectrally queued feature selection, with improved temporal stability, for the purposes of visual odometry. The intended application is future semi-autonomous Unmanned Ground Vehicle (UGV) control as the richness of data sets required to enable human like behavior in these systems has yet to be defined.

Keywords: hyperspectral, LADAR, visual odometry, robotic navigation, autonomous vehicle, sensors

1. INTRODUCTION

1.1 Uses of Autonomous Vehicles

Autonomous vehicles have a wide range of possible applications. In military situations, autonomous vehicles are valued for their ability to keep Soldiers far away from danger. A robot can inspect and disarm a bomb without the need for physical human interaction with explosive materials. Additionally, a UGV can serve as a scout and reconnaissance asset and provide vital operational information about a situation without putting lives on the line. Ideally, as UGVs become more advanced they can perform other dangerous tasks to further remove Soldiers from harms-way. In order for this to be possible further UGV research and development is required in a multitude of areas including navigation, sensing and computation.

Another important application of autonomous systems is in consumer vehicles. One of the most popular new developments is automatic parallel parking. Using sensors, the car is able to undertake this somewhat difficult task on its own. Also, systems exist today that allow vehicles to avoid collision with objects in their environment without human input. This is just a glimpse of what engineers are hoping for in the future.

1.2 Biological Influence

Autonomous vehicles are becoming more of a possibility in our world as technological advances are made and the capabilities of these systems improve. However, in order for autonomous vehicles to operate effectively they require, among other things, a strong set of perceptual systems and a robust set of navigation techniques to intelligently traverse their terrain. While a variety sensory and navigation techniques have been extensively employed [1, 2] there still exist methods utilized by biological systems yet to be fully exploited for robotic navigation (e.g. olfaction, pheromones, etc.)

The mantis shrimp is considered to have the most sophisticated visual system of any species in the animal kingdom [3]. It has the ability to see and discriminate objects in a multitude of wavelengths as well as with acute depth perception. While it is an open research question as to what capabilities this advanced perception system provides, scientists speculate that it can be for communication, hunting, or navigational purposes. This applied research effort was influenced by the potential navigational advantages that such a unique visual system could provide and thus set out to emulate and provide a feasibility analysis.

We have developed a system which integrates a hyperspectral sensor with LADAR to track the movement of an equipped vehicle using a method known as visual odometry. By using the hyperspectral sensor, the vehicle can track objects based on their material, rather than being limited to features such as shape or color.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **23 NOV 2010**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Spectrally Queued Feature Selection for Robotic Visual Odometery

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
David M. Pirozzo; Philip A. Frederick; Shawn Hunt; Bernard Theisen; Mike Del Rose

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA

8. PERFORMING ORGANIZATION REPORT NUMBER
21380RC

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
US Army RDECOM-TARDEC 6501 E 11 Mile Rd Warren, MI 48397-5000, USA

10. SPONSOR/MONITOR’S ACRONYM(S)
TACOM/TARDEC

11. SPONSOR/MONITOR’S REPORT NUMBER(S)
21380RC

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**11**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

1.3 System Summary and Literature Influences

We have developed a new system which can track a vehicle's movement through the use of a hyperspectral sensor and LADAR. Visual odometry is a means of tracking your movement by observing the motion of objects around you.

A key component of visual odometry is tracking features from frame to frame. Many trackable features tend to be corners. A commonly used corner detection algorithm was developed by Harris [4]. The basic idea used in this method locates points where the surrounding neighborhood show edges in more than one direction. Shi and Tomasi [5] later refined this algorithm and claimed that a feature was good as long as the smaller of two eigenvalues was greater than a minimum threshold.

In [6], the authors assumed that a corner looks like a blurred wedge and then computed attributes of the wedge (the amplitude, angle and blur). In [7], the authors generalized that work and proposed calculating corner strength by looking at pixel values within a disc. They calculated the proportion of pixels whose intensity value is with the disc's center, or nucleus. The pixels that have a value closer to the nucleus receive a higher score. They called this measure the USAN, of the Univalue Segment Assimilating Nucleus. If the USAN has a low value, then it is indicative that the USAN is a corner because it is different than its surroundings. These candidates are then run through another test to winnow out bad candidates and the resulting USANS make up the SUSAN, or Smallest USAN.

FAST [8], or Features from Accelerated Segment Test, considers the pixels inside a Bresenham circle (the midpoint circle algorithm) with a radius r, around a candidate point. If there are n contiguous pixels that are all brighter than the nucleus by threshold value t, then the nucleus is considered to be a feature. SIFT [9] (Scale-Invariant Feature Transform) and SURF [10] (Speeded-Up Robust Features) are two types of blob detection algorithms. This class of algorithms detects points or regions in an image that are either brighter or darker than their surroundings. Both SIFT and SURF output a descriptor that is unique for each feature point. Once features have been extracted, the next step is to match them from frame to frame. Common approaches in literature include 2D correlation and Sum of Squared Differences [11] for the corner detectors and nearest neighbor [12] and kd-trees for features that have a descriptor vector.

After features have been extracted and matched, the next step in a visual odometry system is calculate the motion from frame to frame [13]. This step can be troublesome if there are incorrect matches in the features from frame to frame which can be due from the resolution of the camera or if the terrain doesn't have enough distinguishable features. This paper explored the use of a hyperspectral camera to determine if more reliable features can be found.

Using the hyperspectral sensor we can match up targets which relate to one another from one image to the next, and then by using the LADAR data, we are able to calculate the location of each object with respect to the vehicle. By noting these changes in location of multiple objects, we are able to calculate our change in position.

1.4 Other Odometry Sensors/Methods

GPS is commonly used in autonomous systems because it does not accumulate error over time. The drawback however is its reliance on a satellite signal, limiting its application to outdoor use. Stereo vision cameras can be used as an alternative to LADAR. By measuring the disparity between the two camera views, it is possible to roughly determine the distance an object is from the camera. This is faster but less accurate than the LADAR.

There are different methods known as dead reckoning which allow you to track the movement of a vehicle. They usually involve measuring the rotation of the wheels or the speed and travel time of an inertial sensor. When these methods are used on their own, they accumulate error over time, however when combined with other sensors like GPS, they work together to give more accurate results.

2. HARDWARE

2.1 Hyperspectral Imager

Typical cameras record a single intensity value for red, green, and blue light. A hyperspectral imager breaks the spectrum of light into much smaller and more discrete bands in order to form a detailed spectral pattern for every pixel in the image. This process builds a cube of information, each layer of which is composed of a different segment of the spectrum. Every type of material has a slightly different spectral pattern, so by recording a detailed spectral image, we can use this information to determine the material of each object in the scene.

There are many current uses of hyperspectral imaging in various fields of study. Chemical analysis, identifying crop health, biological analysis, target discrimination, material mapping, and mineral deposit analysis are just a few of the many applications which utilize this technology.

2.2 LADAR

LADAR, or Laser Detection and Ranging, traces a laser beam across a single plane, recording distance values at constant angular intervals. By incorporating a tilting system, the entire field of view is able to be scanned. Knowing the current angular interval of the laser, distance to an object returned by the laser, and angle of the tilt system, a map of the scene and the coordinates of any given object is able to be triangulated. LADAR is a commonly used device in autonomous vehicle guidance. Typically it is used in a fixed position and multiple LADAR systems are used to measure each axis. It is useful for obstacle avoidance; however it typically requires the cooperation of other sensors in order to be used for vehicle guidance. Details of both the Hyperspectral Imager and LADAR used in this effort are provided in Table 1.

Table 1. Hardware specs.

Hardware	
<u>Hyperspectral Imager</u>	
Spectral Band	.43-.9 microns
# of Bands	120
Spatial Resolution	640x640
Field of View	5 degrees
Capture Time	about 5 seconds
<u>LADAR</u>	
Angular Resolution	1 to .25 degrees
Range	80 meters
Field of View	180 degrees
Response Time	13 to 53 ms

3. ALGORITHMS

There are three different programs that are used to get from the initial hyperspectral image and LADAR data to the final measurements of vehicle rotation and translation. Our current method for calculating movement is all post processed rather than being done in real time.

3.1 Hyperspectral Analysis

Once a hyperspectral image has been taken, a spectral analysis software suite is used to process the entire hyperspectral cube and output a single image which can then be used by our algorithm to track the motion of the vehicle. The SMACC (Sequential Maximum Angle Convex Cone) algorithm is used first to automatically create a list of spectra which represent various materials found in the scene. This algorithm finds spectra which are the most different from one another in an image. By specifying the number of spectra for it to find, and what tolerance to use when grouping them together, a variety of outputs can be generated which will greatly impact further image processing. With these spectra, a unique spectral library is able to be built that can then be used to compare with other images from the same scene. Another algorithm called SAM (Spectral Angle Mapper) is used to do this comparison. This process compares each pixel of an image to the spectral library that was created in order to group items of the same material together. The benefit of this algorithm is that it is not affected by varying levels of illumination. All items of the same material will be grouped together regardless of how bright or dark they are. When evaluating spectral data, the SAM algorithm measures only the angle of the vector created by a given material and not its length. By doing this, brightness does not affect the results of the calculation. For those pixels that closely match one of the spectra in the library, they are marked with the index value of the matching library spectra. The end result is a single image which is broken into multiple regions, based on the spectral signature (see Fig. 1).

Another approach which can be taken is to manually select spectra from a given environment rather than using the SMACC tool. This allows key elements of a known environment to be identified, providing more information to the robotic system. For instance, given a hyperspectral image, it is possible to manually build a spectral library in which specific types of grass, leaves, bark, stones, pavement, and other materials are identified. This gives the robot further understanding of what things it is seeing, rather than just the ability to match two objects from scene to scene. The drawback of this approach is that it is likely only to operate correctly in the environment where it was developed. New environments would be filled with new types of materials which may not be contained in your library. The SMACC algorithm overcomes this issue by allowing the system to dynamically create a new library based on the current environment, without the need for human guidance.

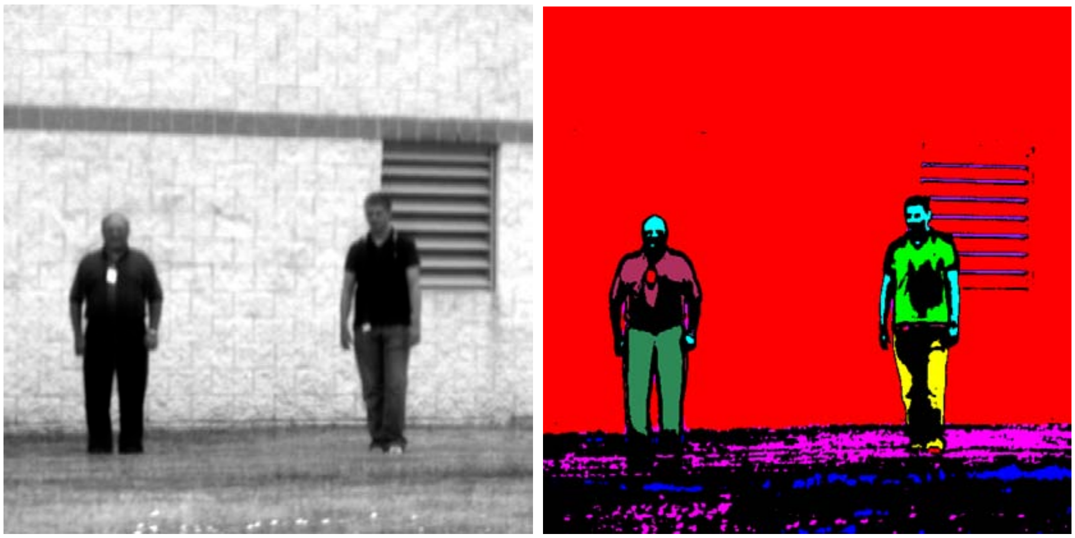

Fig. 1. The image on the left is an image taken with the hyperspectral camera. The image on the right is the same image after it has been processed with hyperspectral analysis software and broken into separate regions using the Spectral Angle Mapper.

3.2 LadarVision

The LadarVision program is designed to be used for a number of LADAR applications. This program was used to align the LADAR data with the hyperspectral image. By changing a number of parameters, such as camera offset, tilt, rotation, and field of view, the two sets of data are able to be aligned. Once this is calibrated, the LADAR data is reformatted to match the resolution of the hyperspectral camera, thereby giving a pixel to pixel correlation. This reformatted LADAR data is then passed on to the tracking algorithm for the final phase of motion detection.

3.3 Tracking Algorithm

In order to track the motion of objects, pairs of images were compared in sequential order. Once feature points were found for each of the images using the previous process, vectors were formed using one point from the first image as the origin and one point from the second image as its destination. For every possible combination of points from the first image to the second, a reference vector was chosen. Each reference vector was then compared with every other possible combination of vectors chosen in the same manner to see if the two vectors were similar. To see if two vectors were similar, they were subtracted, and the difference vector had to be under a certain length to be considered a match. The assumed direction of motion is chosen based on which reference vector had the highest number of matches to other vectors. All the points which match this best vector of motion are then returned for further filtering. The process is displayed in Fig 2.

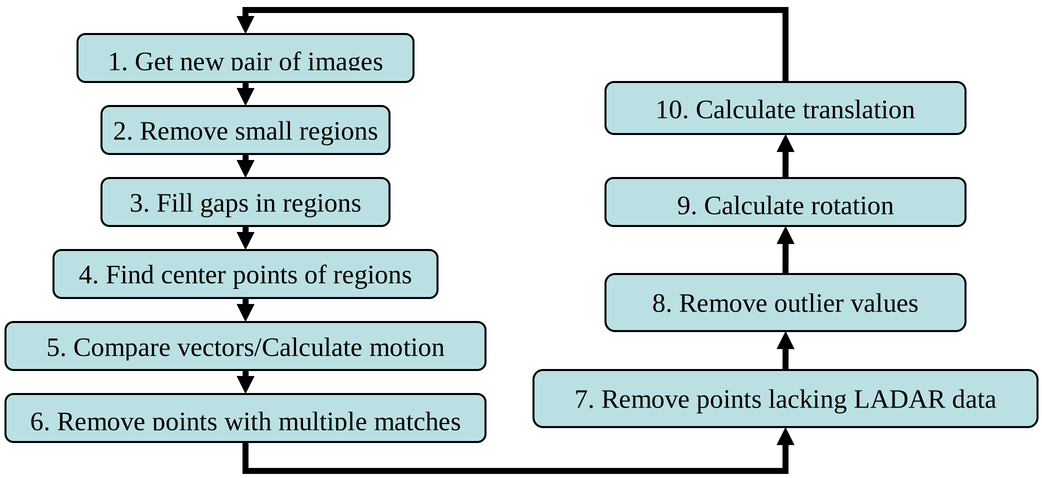

Fig. 2. The figure above shows the process that the tracking algorithm performs on each pair of images in the sequence.

The bulk of the image processing that relates to tracking the motion of the vehicle is done with our algorithm. The first task is to eliminate noisy data that is not worth tracking. Any region whose area falls below a predefined threshold is eliminated. This helps to remove regions which are likely to appear and disappear often. Next, gaps within regions are filled. This helps to keep the region more uniform from one image to the next. The regions which remain after this filtering is complete are assumed to be good tracking targets. Center points of each region are returned for the next phase of the tracking (see Fig 3).

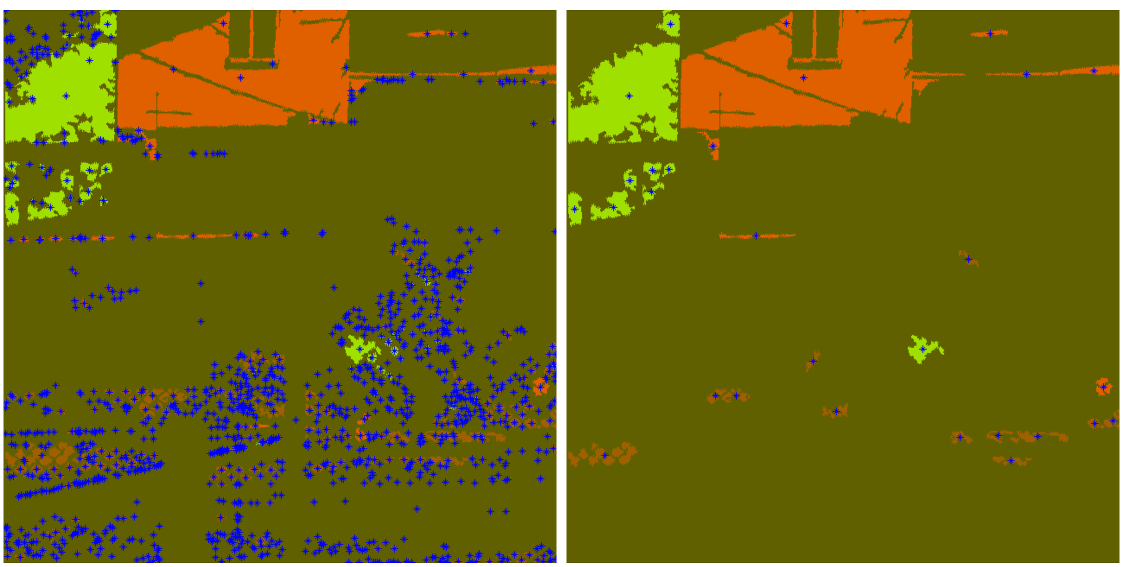

Fig. 3. The left image shows the target points which were selected from each region prior to any filtering, and the right image shows the image after small regions were filtered out.

The first method of filtering is to eliminate any points that had multiple matching vectors coming from them or going to them. An object can only have one true origin point and one true destination point, therefore one of the vectors that matched to that point must be incorrect. It is not possible to reliably determine which of the matching vectors at that point the correct one is, so all of the vectors at that point must be counted as invalid to ensure accurate results (see Fig4). The next method of filtering points is to get rid of any targets which are lacking LADAR data. If either the origin point or the destination point is missing data, they both become useless because a pair of points is needed to determine motion. When recording LADAR data, some objects are beyond the detection range of the device and therefore leave a gap in the data. As well, the slight difference in location of the LADAR sensor and the hyperspectral camera will cause some gaps in the LADAR data when correlating with the hyperspectral image due to their different perspectives. The next step is to filter out values which are too different from the rest of the data. These outliers are removed by finding the median value of all the good data points and checking the points against a certain tolerance value. Once all of this has been done, the final vector is chosen based on points from the material with the highest number of matches (see Fig 5).

All of these filtering techniques are used to help improve the reliability of the data gathered. There are other approaches to performing visual odometry and feature point selection. By combining the information gathered from this technique with that of another such as corner detection, it is possible to increase the reliability of the resulting data. At a most basic level, results from each process could be compared to see whether they agree or not. In a more useful sense, the information could be combined during the feature matching step. The hypserspectral data can be used to check that two features matched by the corner detection approach have matching spectral data, allowing invalid matches to be more quickly detected.

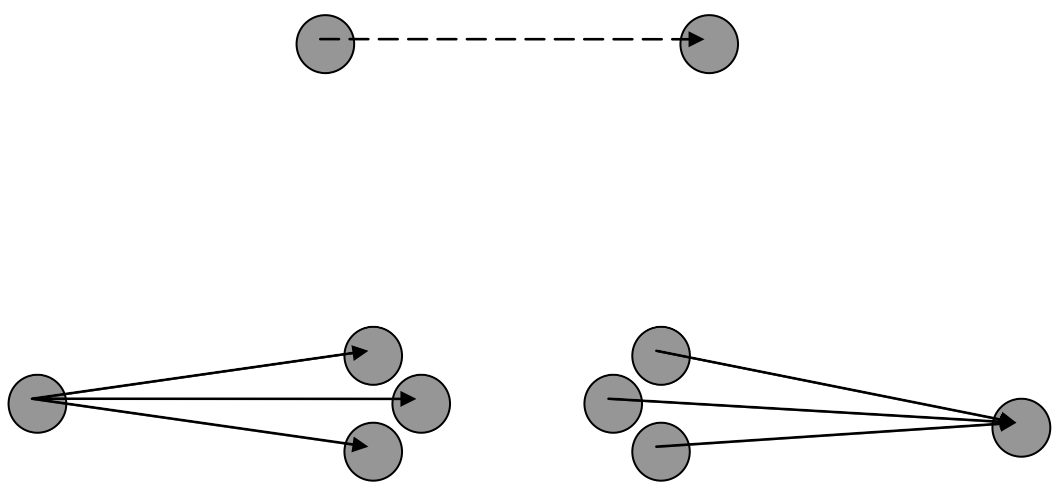

Fig. 4. The above dashed line is a given reference vector being compared to the solid line vectors. On the left, there are multiple vectors originating from the same point, however they all match the reference vector because they are similar enough in size and direction. On the right, there are multiple vectors with the same destination point which all match the reference vector for the same reason. Since only one vector from each set can be correct, they are all discarded to prevent the use of false information in further calculations.

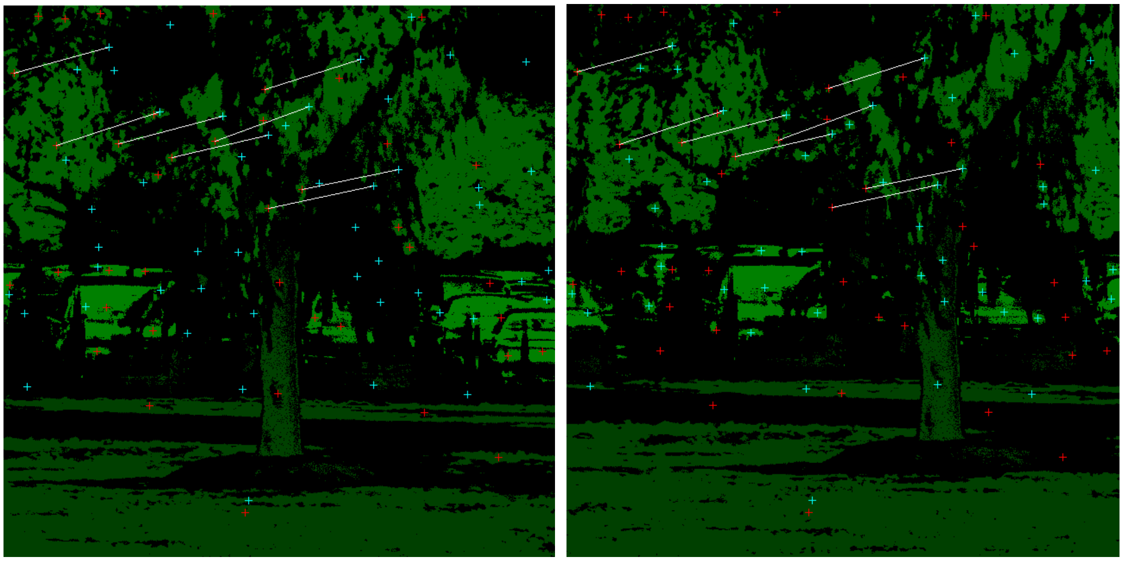

Fig. 5. The image on the left is the first sequentially of a pair of images which are being compared, and the image on the right is the second. Red dots show tracking points from the first image and blue dots show tracking points from the second image. White lines connect pairs of points which are found to be a match using the calculated direction of motion.

3.3.2 Calculating Movement

First, rotation of the vehicle is calculated and to do this the orientation of two points from the first image are compared to the orientation of the same two points in the second image. Take two points which are the furthest distance apart in the first image and make a vector connecting them. Find the same two points in the second image and do the same. Using the depth and horizontal separation of the points, find the angle between the points and the plane of the vehicle for each vector. The difference between these two angles is equivalent to the rotation of the vehicle. With knowledge of the rotation, it is possible to calculate the translation of the vehicle. First, find the angle between one of the target points and the plane of the vehicle in the second position. Add this angle to the vehicle's angle of rotation to determine the angle to the target based on the first position's coordinate axis. Use this angle and the distance to the target to determine the distance in component form with respect to the first coordinate axis. Add these components to the components from the first location to determine the complete translational movement (See Figs 6 & 7).

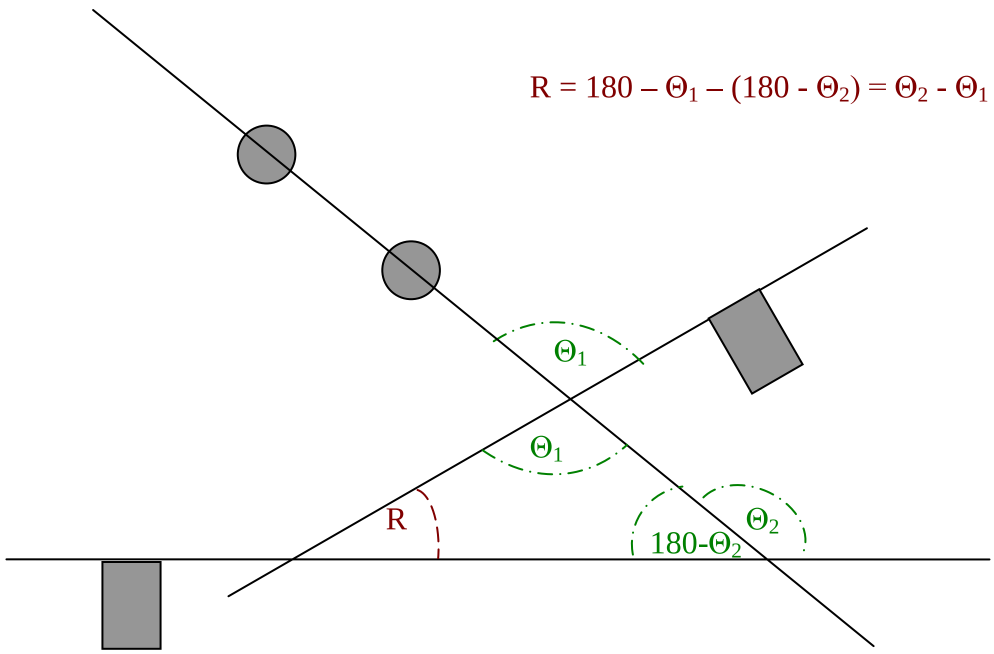

Fig. 6. The figure above shows two vehicle positions and two target positions. The angles Θ_1 and Θ_2 are formed by the plane of the two targets intersecting the plane of the vehicle. Using these angles it is possible to determine the rotation of the vehicle between positions.

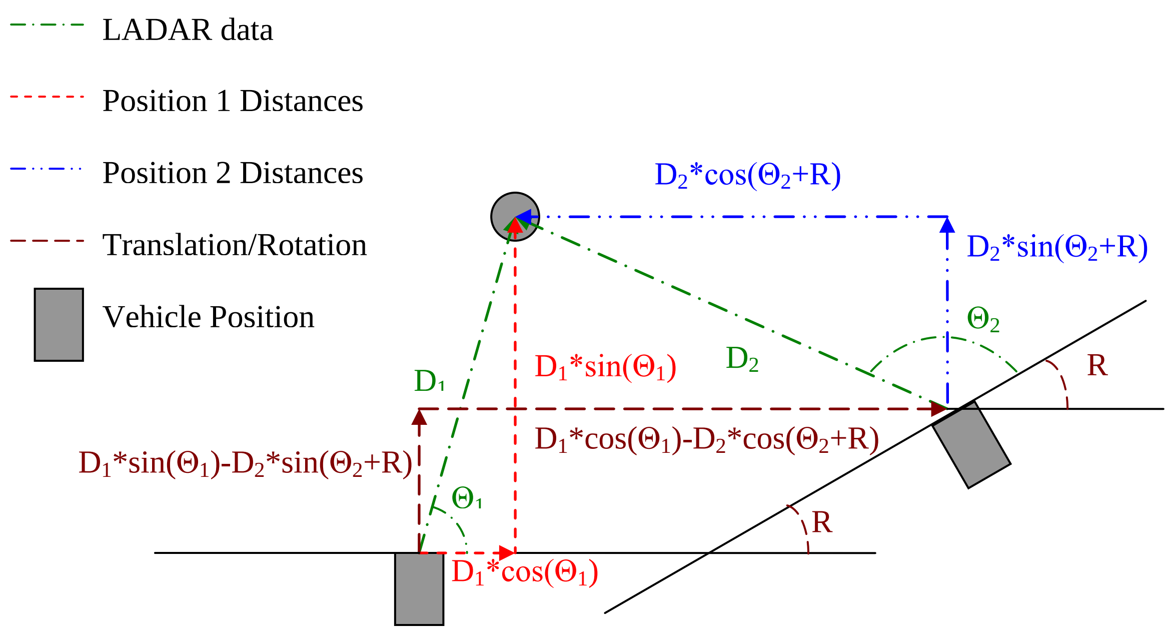

Fig. 7. The figure above shows two vehicle positions and a single target. Using the calculated vehicle rotation and the LADAR distance and angle data, it is possible to calculate the translation of the vehicle between positions.

4. TESTING AND RESULTS

4.1 Method

All of the testing was done on a manually driven vehicle rather than an autonomous platform. Since the hyperspectral camera takes a few seconds to capture an image, the vehicle had to be stationary while taking each image otherwise it would cause too much distortion. The engine vibration of the vehicle also had a noticeable effect on the clarity of images, so the vehicle was turned off while taking images. A number of different locations were tested with various styles of scenery to simulate different situations. For each different setting, a constant distance was measured and the vehicle was moved in a straight line at that increment.

4.2 Observations and Results

It seemed for the most part that either all the images from a given setting would work or none of them would. A common reason for this was due to insufficient LADAR data. The hyperspectral camera has a very narrow field of view, therefore requiring that it be a long distance away from the object. The LADAR, however, has a maximum range of 80 meters, and only seemed effective up to 50, so the objects must be relatively close. The proper balance of the two is required in order to maintain valuable results.

Some environments were more difficult to work with than others. A metal post which doesn't move or change shape is much easier to deal with than trees and bushes which have many small leaves and that move from image to image. Depending on the filtering parameters, this can be difficult to accurately match feature points between images, but after some calibration, these types of objects were able to be reliably tracked as well.

Given that this was simply a plausibility research effort, the process was only designed to measure the accuracy of the longitudinal displacement of the vehicle. There were insufficient resources to accurately measure the lateral movement or the angular rotation; however it should be very small since the vehicle was always moving in a straight line. The calculated angular rotation averaged about 1 degree and the lateral movement about 1 foot when moving a distance of 2 meters forward. It is reasonable to believe that the vehicle could have drifted by these amounts when driven 2 meters, so these values were found to be acceptable for the current level of testing.

4.3 Approach Limitations and Considerations for Future Research

There were a number of complications that we encountered throughout the development process that would need to be dealt with before the system could be applied to an autonomous vehicle. Depending on the type of vehicle, where the system is mounted, and in what way the system is mounted, vehicle vibration may distort the clarity of images taken with the hyperspectral camera. Mounting it further from the engine or having some sort of vibration absorption could be simple solutions to remedy this. A difficulty in making this a widespread application is the extremely high price of hyperspectral cameras. As mentioned before, the narrow field of view of the hyperspectral camera in combination with the short range of the LADAR strictly limits the operating range at which the system will work. If either the LADAR had a further range or the hyperspectral camera had a wider view, this would no longer be an issue. Other hardware may exist that would be more suitable for our application. Above all else, the system is far too slow to be used on an autonomous vehicle. The hyperspectral camera alone takes about 5 seconds to capture an image. After that the target detection and image processing takes at least another 10 seconds. There is no way an autonomous system could make use of the data with that much lag. In order to fix this problem, a faster hyperspectral camera would need to be used. The program may also be improved and made more efficient to help cut down on time. Due to the slow capture time, the vehicle must be stationary when taking an image or else it will turn into a useless blur. If the hyperspectral camera were much faster, this may no longer be an issue. Setting the camera focus and exposure time can be difficult, especially if the lighting conditions are changing frequently. If the exposure is not set correctly, the images are likely to be saturated and no longer usable. A hyperspectral camera may exist with some of these calibration features done automatically.

5. CONCLUSION

This plausibility study has shown that hyperspectral imaging can provide a set of feature data that can assist with feature stability for visual odometry. However the current hardware complications must be overcome before it could be used in an actual robotic application. The data provided by this type of sensor is quite unique and is not typically available through the use of other devices. This gives the robotic system the ability to perceive new aspects of the environment which complements data provided by other sensors. By giving the system this new set of data, it is possible to develop more robust solutions and improve performance.

REFERENCES

1. J. Borenstein, H. R. Everett, L. Feng, and D.Wehe, “Mobile robot positioning: Sensors and techniques,” J. Robot. Syst., vol. 14, pp. 231–249, Aug. 1997.
2. W. Burgard, D. Fox, and S. Thrun. Active mobile robot localization. In Proceedings of the International Joint Conference on Articial Intelligence (IJCAI), Nagoya, Japan, August 1997.
3. Thomas W.Cronin and Justin Marshall, "Parallel Processing and Image Analysis in the Eyes of Mantis Shrimps," The Biological Bulletin 200, 177-189 (2001).
4. Harris, C. and M. Stephens. A combined corner and edge detector. 1988: Manchester, UK.
5. Shi, J. and C. Tomasi. Good features to track. 2002: IEEE.
6. Guiducci, A., Corner characterization by differential geometry techniques. Pattern Recognition Letters, 1988. 8(5): p. 311-318.
7. Smith, S. and J. Brady, SUSAN—A new approach to low level image processing. International journal of computer vision, 1997. 23(1): p. 45-78.
8. Rosten, E. and T. Drummond, Machine learning for high-speed corner detection. Computer Vision–ECCV 2006, 2006: p. 430-443.
9. Lowe, D. Object recognition from local scale-invariant features. 1999: Published by the IEEE Computer Society.
10. Bay, H., T. Tuytelaars, and L. Van Gool, Surf: Speeded up robust features. Computer Vision–ECCV 2006, 2006: p. 404-417.
11. Bradski, G. and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library. 2008: O'Reilly Media.
12. Beis, J. and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. 2002: IEEE.
13. Hartley, R. and A. Zisserman, Multiple view geometry in computer vision. 2003: Cambridge Univ Pr.